

Abstract:
DE 29823119 U

NOVELTY - The automation device (1) is provided by a hyper text transport protocol server, which is coupled via an internet and/or intranet network (2) with a visualisation device (3) for graphical visualisation of the controlled technical process using hyper text markup language pages. The automation device provides applets for dynamic components of the technical process, loaded as components of the hyper text markup language pages.

USE - For technical process automation device.

ADVANTAGE - Allows visualisation of the controlled technical process at a remote point.

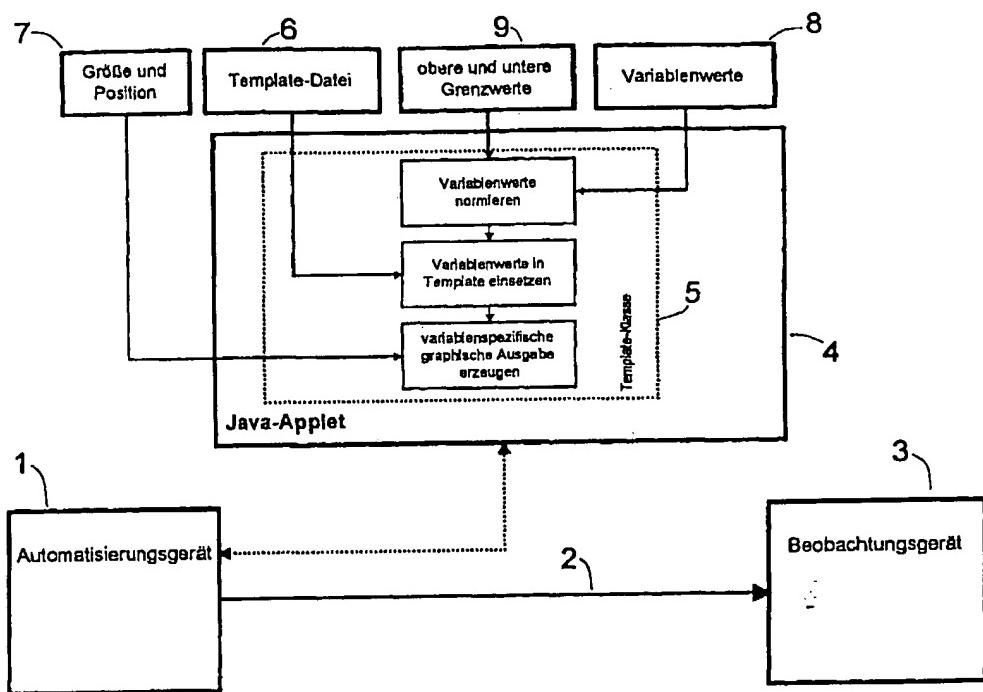
DETAILS OF DRAWING - The drawing shows the automation device coupled to a display.

1 Automisation device

2 Internet/intranet network

3 Display

Dwg.1/1





⑯ BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑯ **Gebrauchsmuster**
⑯ **DE 298 23 119 U 1**

⑯ Int. Cl.⁶:
G 05 B 15/02
G 07 C 3/12

DE 298 23 119 U 1

⑯ Aktenzeichen: 298 23 119.0
⑯ Anmeldetag: 28. 12. 98
⑯ Eintragungstag: 25. 2. 99
⑯ Bekanntmachung im Patentblatt: 8. 4. 99

⑯ Inhaber:
Siemens AG, 80333 München, DE

Rechercheantrag gem. § 7 Abs. 1 GbmG ist gestellt

⑯ Automatisierungsgerät

DE 298 23 119 U 1

20.12.98

Beschreibung

Automatisierungsgerät

- 5 Die Erfindung betrifft ein Automatisierungsgerät zur Steue-
rung eines technischen Prozesses, welches über ein Netzwerk
in Form eines Internet- und/oder Intranet-Netzwerkes mit
einem Beobachtungsgerät verbindbar ist, welches zur Prozeß-
beobachtung vorgesehen ist.
- 10
- In der Automatisierungstechnik erfolgt gewöhnlich die Dar-
stellung von Meßgrößen, Zustandsgrößen und Rechenergebnissen
an der Benutzerschnittstelle (Human/Machine Interface) vor-
wiegend in grafischer Form. Z. B. wird der Füllstand eines
- 15 Kessels dargestellt durch eine realitätsnahe Darstellung von
Kessel und Inhalt, wobei die Darstellung des Kessels immer
konstant ist, die Darstellung des Inhalts jedoch von einem
zuvor erfaßten Meßwert abgeleitet wird. Neben vorgefertigten
grafischen Elementen sind auch für anlagenspezifische Auto-
- 20 matisierungsfunktionen eigens konstruierte grafische Dar-
stellungen vorgesehen.

Ein Automatisierungsgerät der eingangs genannten Art ist aus
der WO 97/26587 bekannt. Zwei Fertigungsstandorte eines
25 global verteilten Automatisierungsverbundes sind über ein an
sich bekanntes globales Netzwerk „Internet“ miteinander ver-
bunden, wobei diese Standorte mehrere Geräte in Form von
Automatisierungsgeräten, Programmiergeräten, Servern, Bedien-
und Beobachtungsgeräten und Workstations umfassen. Eine
30 Prozeßbedienung und Prozeßführung der zu steuernden Prozesse
in den Fertigungsstandorten erfolgt durch ein Bedien- und
Beobachtungsgerät. Dieses erzeugt ein Bedien- und Beobach-
tungs-Softwarebausteine umfassendes Bedien- und Beobachtungs-
programm zur Erstellung und Darstellung eines mehrere Bild-
35 objekte umfassenden Prozeßbildes, wobei die Bildobjekte zu
Software-Funktionsbausteinen des Steuerprogramms im Automati-
sierungsgerät in Beziehung (in Wechselwirkung) stehen. Die

28.12.98

Bedien- und Beobachtungs-Softwarebausteine sind objekt-orientiert ausgebildet und direkt über das Internet übertragbar.

5 Aus den Web-Seiten

<http://www.wonderware.de/produkte/fsuite.htm> der Firma Wonderware ist ein Manufacturing Management Information System (MMI-System) bekannt, welches Mittel und Maßnahmen zur Visualisierung und zum Überwachen von Produktionsabläufen über das Internet/Intranet ermöglicht.

Der vorliegenden Erfindung liegt die Aufgabe zugrunde, ein Automatisierungsgerät der eingangs genannten Art zu schaffen, mit welchem eine verbesserte Beobachtung eines zu steuernden 15 technischen Prozesses ermöglicht wird.

Diese Aufgabe wird durch die im kennzeichnenden Teil des Anspruchs 1 angegebenen Maßnahmen gelöst.

20 HTTP-Server sind an sich bekannt. Diese unterstützen ein sogenanntes „Hyper Text Transport Protokoll“, welches im World Wide Web von verteilten Hypermedia-Systemen als Such- und Übertragungsprotokoll verwendet wird, um auf verteilte Objekte zuzugreifen. Durch dieses Protokoll ist es möglich, 25 HTML-Seiten („Hyper Text Markup Language“-Seiten) schnell zu übertragen, welche durch geeignete HTML-Browser darstellbar sind. Ferner sind in diese HTML-Seiten sogenannte Applets einbindbar, welche einen durch einen HTML-Browser interpretierbaren und durch diesen Browser ausführbaren Bytecode 30 umfassen.

Vorteilhaft ist, daß eine dynamische Komponente, z. B. eine Komponente in Form eines grafisch darzustellenden Kessel-Füllstandes, dynamisch darstellbar ist. Dabei wird für im wesentlichen statische Komponenten, welche sich nie oder 35 selten ändern, z. B. für auszutauschende Anlagenteile, die entsprechende darzustellende HTML-Seite neu geladen. Für sich

28.12.98

3

häufig ändernde Komponenten, z. B. für Werte von Prozeßvariablen, werden die entsprechenden Applets im Automatisierungsgerät erzeugt, in das Beobachtungsgerät geladen und durch dieses ausgeführt.

5

Das als HTTP-Server ausgebildete Automatisierungsgerät generiert dynamisch grafische Prozeßvariablen derart, daß in gängigen HTML-Browsern (z. B. Netscape, Internet-Explorer) und HTML-Editoren für spezifische Automatisierungsanlagen 10 entsprechende Informations- und Bedienoberflächen einfach erstellbar sind.

Weitere vorteilhafte Ausgestaltungen der Erfindung ergeben sich aus den in den Unteransprüchen angegebenen Maßnahmen.

15

Die Erfindung wird nachfolgend anhand eines Ausführungsbeispiels unter Bezugnahme auf die einzige Figur der Zeichnung näher erläutert. Die Figur der Zeichnung zeigt in schematischer Darstellung die Generierung einer HTML-Seite.

20

Ein als HTTP-Server ausgebildetes Automatisierungsgerät 1 ist über ein Netzwerk 2 in Form eines an sich bekannten Internet- und/oder Intranet-Netzwerkes mit einem Beobachtungsgerät 3 verbindbar, welches zur Prozeßbeobachtung eines durch das 25 Automatisierungsgerät 1 zu steuernden technischen Prozesses vorgesehen ist. Das Beobachtungsgerät 3 ist als HTML-Browser ausgebildet, welchem über das Netzwerk 2 HTML-Seiten zur Aufbereitung und grafischen Darstellung des technischen Prozesses zuführbar sind. Eine Änderung von im wesentlichen statischen Komponenten des technischen Prozesses zeigt das Automatisierungsgerät 1 dem Beobachtungsgerät 3 an, wodurch 30 dieses die entsprechende HTML-Seite aus dem Automatisierungsgerät 1 neu lädt und darstellt. Dadurch sind z. B. Umkonfigurationen einer technischen Anlage darstellbar.

35

Für den Fall, daß auch dynamische Komponenten des technischen Prozesses (z. B. Werte von Meßgrößen) innerhalb einer HTML-

20.12.98

Seite grafisch darzustellen sind, erzeugt das Automatisierungsgerät sogenannte Applets, welche in das Beobachtungsgerät zur Darstellung der HTML-Seite ladbar und durch das Beobachtungsgerät ausführbar sind. Im Ausführungsbeispiel ist 5 ein Java-Applet 4 dargestellt, welches die darzustellenden grafischen Prozeßvariablen (dynamische Komponenten) durch im HTML-Browser (Beobachtungsgerät 3) ablaufende Java-Applets erzeugt. Das Java-Applet 4 verwendet eine sogenannte Template-Klasse 5. Diese Klasse erhält als Initialisierungs-10 parameter den Namen einer ASCII-Datei, die eine Reihe von parametrierbaren Grafikprimitiven (z. B. Kreis, Rechteck, etc.), Farbdefinitionen oder den Aufruf einer weiteren Template-Datei 6 enthält. Die Parameter der Grafikprimitive können neben festen Werten auch arithmetische Ausdrücke sein, 15 die einen oder mehrere Prozeßvariablenwert(e) enthalten. Größe und Position der dargestellten Grafikelemente sind in einer Elementen-Datei 7 hinterlegt und können sich in Abhängigkeit von einem oder mehreren Prozeßwerten verändern. Eingabeparameter einer Template-Instanz, welche in einer 20 Variablenwerte-Datei 8 hinterlegt sind, sind die oder der momentane(n) Variablenwert(e) in Form eines Arrays aus Fließkommawerten in doppelter Genauigkeit oder eines Arrays aus Strings, die die Zahlenwerte als ASCII-Text enthalten. Außerdem werden für jeden Variablenwert ein oberer und unterer Grenzwert angegeben, welche in einer Grenzwerte-Datei 9 25 hinterlegt sind, so daß die Werte, die in einem arithmetischen Ausdruck eines Grafikelementes verwendet werden, stets auf einen Bereich zwischen 0 und 999 normiert werden können. Die normierten Variablenwerte werden in das Template eingesetzt, d. h., die arithmetischen Ausdrücke, die Variablenwerte enthalten, werden ausgewertet und die jeweils entstehenden Grafikelemente dargestellt. Dieser Vorgang wird vorzugsweise zyklisch wiederholt und die dargestellten 30 Grafikelemente den jeweils aktuellen Variablenwerten angepaßt. 35 Um in einem Java-Applet mehrere Templates oder zusätzliche Informationen, die die Templates ergänzen, darstellen zu

20.12.90

5

können, besteht die Möglichkeit, die Größe und Position eines Templates als Ganzes innerhalb der dargestellten Applet-Fläche oder innerhalb eines aufrufenden Templates festzulegen.

5

Ein Beispiel für die Generierung von grafischen Prozeßvariablen ist in der folgenden Tabelle dargestellt.

28.12.90

Syntax der Template-Datei

Die Syntax einer Template-Datei ist in Backus-Naur-Form wie folgt beschrieben:

```
<Template>      ::= { <Line> <Terminator> [ <CRLF> ] }

<Line>          ::= <VarDecl> | <Comment> | <Command>

<VarDecl>       ::= „var“ { <Separator> <Variable> }

<Comment>        ::= „rem“ { <Character> | <Quote> | <Blank> }

<Command>        ::= <Keyword> <Parameterliste>

<Keyword>        ::= „image“ | „oval“ | „line“ | „box“ | „rbox“ | „arc“ |
                      „color“ | „text“ | „font“ | „bkgrnd“

<Parameterliste> ::= <Separator> <Parameter> { <Separator> <Parameter> }

<Parameter>      ::= <String> | <Modifier> | <Expression>

<Expression>     ::= arithmetischer Ausdruck

<Variable>       ::= Variablenbezeichner nach Step7 Konvention

<Modifier>        ::= <Character>

<String>          ::= <Quote> { <Character> | ; | <Blank> | „\“ <Quote> }

<Quote>

<Character>      ::= ASCII 33 bis ASCII 126 außer ASCII 34 und ASCII 59

<Quote>          ::= Hochkomma (ASCII 34)

<Separator>      ::= <Blank>

<Blank>          ::= Leerzeichen (ASCII 32)

<Terminator>    ::= „;“ (ASCII 59)

<CRLF>          ::= <CR> [ <LF> ]

<CR>            ::= Carriage Return (ASCII 13)

<LF>            ::= Line Feed (ASCII 10)
```

28.12.98

Für den Typ <Expression> gelten die Syntaxregeln für arithmetische Ausdrücke.

Parameter der Keywords

Koordinaten- und Größenangaben sind im folgenden kursiv dargestellt (z. B. *x_start*). Die Werte liegen stets in einem Bereich zwischen 0 und 999 und sind als relative Angaben zu verstehen. Dadurch ist die Skalierbarkeit der grafischen Darstellung gewährleistet. Der absolute Wert x bzw. y errechnet sich zu $x \cdot \text{WIDTH}/1000$ bzw. $y \cdot \text{HEIGHT}/1000$ mit WIDTH als Applet-Breite und HEIGHT als Applet-Höhe.

Zeichenketten sind stets mit Hochkomma eingefasst. Falls eine Zeichenkette ein Hochkomma enthalten soll, muß ein Backslash "\ vorangestellt werden.

Ein Kommentar beginnt mit dem Schlüsselwort **rem**, kann anschließend beliebige Zeichen (außer ";") enthalten und endet mit dem Terminator-Zeichen ";".

Die in arithmetischen Ausdrücken verwendeten Variablen müssen (vorzugsweise zu Beginn der Template-Datei) mit Hilfe der Anweisung **var** deklariert werden.

image filename

Verwendet die Pixelgrafik aus der Datei mit dem Namen filename als Hintergrund für das Applet. Der Wert ist vom Typ <String>, und es gelten die Syntaxregeln für Dateinamen. Die Pixelgrafik kann alle Grafikelemente enthalten, die durch Variablenwerte nicht verändert werden.

bkgnd red_intensity green_intensity blue_intensity

Legt die Hintergrundfarbe des Applets fest; die Werte liegen im Bereich zwischen 0 und 255. Die Werte sind

26.12.90

vom Typ <Expression>.

color `red_intensity green_intensity blue_intensity`

Legt die Farbe für alle folgenden grafischen Operationen fest; die Werte liegen im Bereich zwischen 0 und 255. Die Werte sind vom Typ <Expression>.

line `x_start y_start x_end y_end`

Zeichnet eine Linie zwischen den beiden Punkten (`x_start, y_start`) und (`x_end, y_end`). Die Werte sind vom Typ <Expression>.

oval `modifier x_start y_start x_end y_end`

Zeichnet eine Ellipse, deren umschreibendes Rechteck durch die zwei Koordinaten (`x_start, y_start`) und (`x_end, y_end`) gegeben ist. Die Werte sind vom Typ <Expression>. Für `modifier` gleich „d“ wird die Ellipsenkontur gezeichnet; für `modifier` gleich „f“ wird die Ellipse ausgefüllt.

box `modifier x_start y_start x_end y_end`

Zeichnet ein Rechteck, das durch die zwei Koordinaten (`x_start, y_start`) und (`x_end, y_end`) gegeben ist. Die Werte sind vom Typ <Expression>. Für `modifier` gleich „d“ wird die Rechteckkontur gezeichnet; für `modifier` gleich „f“ wird das Rechteck ausgefüllt.

rbox `modifier x_start y_start width height arc_width
arc_height`

Zeichnet ein Rechteck mit abgerundeten Ecken, das durch die zwei Koordinaten (`x_start, y_start`) und (`x_end, y_end`) gegeben ist; der horizontale (vertikale) Durchmesser des Bogens an allen vier Ecken ist durch `arc_width` (`arc_height`) gegeben. Die Werte sind vom Typ <Expression>. Für `modifier` gleich „d“ wird die

28.12.98

Rechteckkontur gezeichnet; für modifier gleich „f“ wird das Rechteck ausgefüllt.

arc modifier *x y width height start_angle arc_angle*

Zeichnet einen Kreisbogen, dessen umschreibendes Rechteck durch die zwei Koordinaten (*x_start*, *y_start*) und (*x_end*, *y_end*) gegeben ist. Der Startwinkel ist durch *start_angle* und der Endwinkel durch *end_angle* gegeben. Die 0-Grad-Marke liegt dabei an der „drei Uhr“-Position; der Maximalwert für *start_angle* und *arc_angle* liegt bei 360 Grad. Die Werte sind vom Typ <Expression>. Für modifier gleich „d“ wird die Kreisbogenkontur gezeichnet; für modifier gleich „f“ wird der Kreisbogen ausgefüllt.

font name style size

Legt den Namen, den Stil und die Größe aller folgenden Textausgaben fest. Als name sind momentan folgende Strings möglich: Serif, SansSerif, Monospaced. Als style sind folgende Strings möglich: bold, italic, bold_italic, plain. Die Größe size ist als relative Größe zur Applet-Höhe anzugeben.

text *x y Zeichenkette | Variable { Zeichenkette | Variable}*

Gibt einen Text, gegeben durch eine oder mehrere Zeichenkette(n) oder Variablenname(n), an der Position (*x*, *y*) aus.

28.12.98

10

Schutzansprüche

1. Automatisierungsgerät zur Steuerung eines technischen Prozesses, welches über ein Netzwerk (2) in Form eines Internet- und/oder Intranet-Netzwerkes mit einem Beobachtungsgerät (3) verbindbar ist, welches zur Prozeßbeobachtung vorgesehen ist,
dadurch gekennzeichnet, daß
 - das Automatisierungsgerät (1) als HTTP-Server ausgebildet ist, wobei dem Beobachtungsgerät (3) durch den HTTP-Server (1) HTML-Seiten zur Aufbereitung und grafischen Darstellung des technischen Prozesses zuführbar sind,
 - das Automatisierungsgerät (1) für dynamische Komponenten des technischen Prozesses Applets (4) erzeugt, welche als Bestandteile von HTML-Seiten in das Beobachtungsgerät (3) ladbar und im Beobachtungsgerät (3) ausführbar sind, und
 - das Automatisierungsgerät (1) eine Änderung einer im wesentlichen statischen Komponenten des technischen Prozesses dem Beobachtungsgerät (3) anzeigt, wodurch das Beobachtungsgerät die HTML-Seite neu lädt.
2. Automatisierungsgerät nach Anspruch 1, dadurch gekennzeichnet, daß im Automatisierungsgerät (1) in HTML-Seiten eingebettete Java-Applets (4) in das Beobachtungsgerät (3) ladbar sind, welche die Generierung grafischer Prozeßvariablen ermöglichen, wobei die grafischen Prozeßvariablen durch im Beobachtungsgerät (3) ablaufende Java-Applets generiert werden.
3. Automatisierungsgerät nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß das Automatisierungsgerät (1) die Applets (4) zyklisch erzeugt.

28.12.96

1/1

